



CERTIFICATE OF MAILING BY "FIRST CLASS MAIL"

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Mail Stop: Appeal Brief – Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on Thursday December 29, 2005.

Ali Makoui

Ali Makoui

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In the application of:

Scott Ryder

Serial No.: 09/751,903

Filing Date: 12/28/2000

For: METHOD AND APPARATUS FOR
AUTOMATIC REMOTE VOLUME
MOUNTING

Examiner: Kang, Paul H

Group Art Unit: 2141

TRANSMITTAL LETTER FOR FILING AN APPEAL BRIEF

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Examiner:

In complete response to the Notice of Appeal filed on 8/29/2005, attached please find the following mailed on 12/29/2005:

1. An Appeal Brief;
2. A Return Receipt Post Card;
3. A Petition for Extension of Time;
4. A Credit Card Payment Form; and
5. An Amendment to cancel a claim.

The fee has been calculated as follows:

FOR	NUMBER	NUMBER OVER ALLOTMENT	RATE	CALCULATIONS
ADDED CLAIMS			x \$50.00	
ADDED INDEPENDENT CLAIMS			x \$200.00	
MULTIPLE DEPENDENT CLAIM(S) (if applicable)			+ \$360.00	
FEE FOR FILING A BRIEF IN SUPPORT OF AN APPEAL:				\$500.00
Total from Above				\$500.00
Reduction by 1/2 for filing by small entity (Note 37 C.F.R. §§ 1.9, 1.27, 1.28). The Applicant hereby states that it qualifies as a small entity under 37 CFR 1.27				\$0.00
				TOTAL = \$500.00

The Commissioner is hereby authorized to charge any additional fees under 37 C.F.R. §§ 1.16 and 1.17 that may be required by this transmittal and associated documents, or to credit any overpayment to **Deposit Account No. 50-1128**.

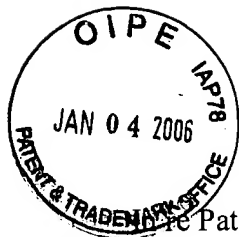
Respectfully submitted,

Dated: December 29, 2005

By:

Ali Makoui
Ali Makoui
Registration No. 45,536

Stattler Johansen & Adeli LLP
1875 Century Park East, Ste 1050
Los Angeles, CA 90067
Tel: (310) 785-0140x302
FAX: (310) 785-9558



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Patent Application for:

Scott Ryder

Serial No.: 19/751,903

Filing Date: 12/28/2000

For: **METHOD AND APPARATUS FOR
AUTOMATED REMOTE VOLUME
MOUNTING**

Examiner: Kang, Paul H.

Group Art Unit: 2141

APPEAL BRIEF

COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This is an Appeal from the final rejection of claims 1-24 in the above-referenced application. In accordance with 37 C.F.R. § 41.37, this Brief is filed along with the accompanying Appendices and the required fee. Please charge any additional fees or credit any overpayment to Deposit Account No. 501128.

I. REAL PARTY IN INTEREST

The real party in interest to this Appeal is Apple Computer, Inc., a California Corporation, having its principal place of business in Cupertino, California.

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences known to Appellants, the Appellants' legal representative, or assignees thereof.

AF
JW

III. STATUS OF CLAIMS

Claims 1-24 are pending in the present application. The Examiner has rejected claims 1-24. Appellants have canceled claim 22 in a concurrently submitted amendment and hereby appeal the rejection of claims 1-21 and 23-24.

IV. STATUS OF AMENDMENTS

Appellants have submitted an amendment concurrent with this appeal brief to cancel claim 22.

V. SUMMARY OF CLAIMED SUBJECT MATTER

1. Independent Claims

A. Claim 1

Claim 1 recites a method that automatically mounts several remote volumes to a client. Without receiving a request from a user of the client to mount the volumes, the method requests a first server to mount the volumes. *See, e.g.*, page 6, lines 23-27. The method returns a set of mounting parameters from the first server. *See, e.g.*, page 5, line 26 to page 6, line 7; page 9, lines 14-22; and Figure 2, block 250. The method automatically mounts the volumes using the returned parameters. *See, e.g.*, page 15, lines 5-8 and Figure 5, block 550. After mounting, the volumes behave as native to the client, even though the volumes are located remote to the client. *See, e.g.*, page 6, line 27 to page 7, line 2.

B. Claim 19

Claim 19 recites a system for automatically mounting volumes over a network. *See, e.g.*, page 7, line 27 to page 8, line 14. The system has a client configured to automatically request the mounting of the volumes without the need for a user of the client to request the mounting of the volumes. *See, e.g.*, page 5, lines 3-9 and Figure 1, elements 100 and 104. The system also has a server coupled to the client. *See, e.g.*, page 1, lines 9-11 and Figure 1, element 150. The server is configured to return a mounting command block to the client. *See, e.g.*, page 5, line 20 to page 6, line 3.

D. Claim 23

Claim 23 recites an apparatus in a computer system that has a processor and memory. The apparatus automatically mounts several remote volumes to the computer system. *See, e.g.*, page 15, line 12 to page 16, line 9; and Figure 6. The apparatus includes a plug-in module configured to receive from a server a mounting command block. *See, e.g.*, page 5, lines 9-11; page 5, line 26 to page 2, line; and page 17, lines 13-15. The mounting command block includes volume-mounting parameters. *See, e.g.*, page 5, line 26 to page 6, line 7. The computer system also includes an application programming interface coupled to the plug-in module and configured to automatically mount the volumes specified in the mounting command block. *See, e.g.*, page 15, lines 5-11. The mounting is performed without the need for user intervention.

E. Claim 24

Claim 24 recites an apparatus for automatically mounting several volumes to a client. The apparatus includes means for requesting of a first server the mounting of the volumes without receiving a request from a user of the client to mount the volumes. *See, e.g.*, page 6, lines 23-27. The

apparatus includes means for returning by the first server a set of mounting parameters. *See, e.g.*, page 5, line 26 to page 6, line 7; page 9, lines 14-22; and Figure 2, block 250. The apparatus also includes means for automatically mounting the volumes using the returned parameters. *See, e.g.*, page 15, lines 5-8 and Figure 5, block 550. After mounting, the volumes behave as if native to the client, even though the volumes are located remote to the client. *See, e.g.*, page 6, line 27 to page 7, line 2.

2. Dependent Claims

A. Claim 5

Claim 5 recites the method of claim 1 to further include detecting the presence of a cookie on the client. The cookie indicates by its presence that the client has installed a plug-in module. *See, e.g.*, page 5, lines 11-15 and page 8, line 27 to page 9, line 11. If the cookie is not detected, the volume mounting is terminated.

B. Claim 6

Claim 6 is indirectly dependent to claim 1. Claim 6 recites that returning the set of mounting parameters by the first server includes, if successfully authenticated, retrieving the parameters from a profile database on the first server. The parameters are indexed by the login ID. The method assembles a mounting command block that consists of a mount command and the retrieved parameters. *See, e.g.*, page 9, lines 11-25.

D. Claim 8

Claim 8 is indirectly dependent to claim 1. Claim 8 recites that returning the set of mounting

parameters include retrieving the parameters from a profile datastore on the first server. The parameters are indexed by an identifier for the event. A mounting command block is assembled which consists of a mount command and the retrieved parameters. *See, e.g.,* page 9, lines 14-25.

E. Claim 9

Claim 9 is dependent on claim 1. Claim 9 recites that the first server returns parameters via a plug-in module that is installed on the client. *See, e.g.,* page 5, lines 9-11; page 5, line 26 to page 2, line; and page 17, lines 13-15.

F. Claim 10

Claim 10 is dependent on claim 9. Claim 10 recites that automatically mounting includes obtaining an address from the first server. The method compares the address for the first server against a built-in list of approved servers. If the address is not contained on the built-in list, the method terminates the process of automatically mounting. *See, e.g.,* page 6, lines 11-22.

G. Claim 11

Claim 11 is dependent on claim 10. Claim 11 recites that if the address obtained from the server is contained on the built-in list, then the method passes the parameters by the plug-in to an application programming interface (API) of the client. The method completes the mounting of the volume by the API. The mounting is achieved absent of user intervention. *See, e.g.,* page 15, lines 5-11.

H. Claim 21

Claim 21 is indirectly dependent on claim 19. Claim 21 recites that the system further

includes an application programming interface (API) coupled to the plug-in and configured to automatically mount the volumes specified in the mounting command block without the need for user intervention. *See, e.g.,* page 15, lines 5-11.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

I. Claims 1-22 and 23-24 are rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent. 6,212,640 to Abdelnur et al. ("Abdelnur") in view of U.S. Patent 6,553,415 to Manwiller ("Manwiller").

VII. ARGUMENT

The Examiner erred in rejecting the claimed invention by misapplying standards under 35 U.S.C. §103(a).

A. The Subject Claims 1, 19, and 24 are Patentable under 35 U.S.C. §103(a) over Abdelnur in View of Manwiller

In rejecting the subject claims 1 under §103, the Examiner stated the following:

Abdelnur discloses the invention substantially as claimed. Abdelnur discloses requesting of a first server said mounting of said volumes; returning by said first server a set of mounting parameters; and automatically mounting said plurality of volumes utilizing said returned parameters, said parameters, said volumes located remote to said client, said volumes after mounting behaving as would a volume native to said client, (*See* Column 5 Lines 61-67 & Column 6 Lines 1-15).

However, Abdelnur does not explicitly teach the method comprising mounting without receiving a request from a user of said client to mount said plurality of remote volumes. In the same field of endeavor, Manwiller teaches a system and method for automatically establishing connection with a remote device over a distributed network using API's (Manwiller, col. 5, lines 20-55)

(Office Action mailed March, 25, 2005, pages 2-3)

1. **The Cited References Do Not Disclose, Teach, or Suggest Returning a Set of Mounting Parameters from a Server and Automatically Mounting a Set of Volumes Utilizing the Returned Parameters**

Appellant respectfully submits that the cited references do not disclose, teach, or even suggest a method that (1) returns a set of mounting parameters from a server and (2) automatically mounts a set of volumes by utilizing the returned parameters. Rather, Abdelnur discloses that a remote server mounts a volume, generates a set of results, and returns the generated results to the client.

Specifically, the cited paragraphs of Abdelnur disclose a Network File System (NFS).
Abdelnur discloses:

An NFS server is a computer that shares its resources with other computers (NFS clients) on the network, using the NFS service. *See*, Abdelnur, column 6, lines 4- 6.

Using NFS, a resource (i.e., software) physically linked to a NFS server may be “NFS mounted.” The resource that is “NFS mounted” is accessible to all NFS clients as if the software were stored locally at the client. *See*, Abdelnur, column 6, lines 7-10.

Appellant respectfully submit that in Abdelnur the NFS mounts the volumes. The NFS system is used by an application to make a request for a procedure to be performed and the results to be returned. Specifically, Abdelnur discloses:

Using NFS, application makes for a procedure to be performed, (e.g., a read or write request). *See*, Abdelnur column 6, lines 33-37.

While application is waiting for a response to its request, NFS client encodes the contents of the local request into a remote-procedure-call (RPC). *See*, Abdelnur column 6, lines 40-42.

Once NFS server receives a RPC request from NFS client, the request is decoded and processed as a local file system operation. The result generated by NFS server is encoded and returned to NFS client. (Emphasis added). *See*, Abdelnur, column 6, lines 53-56.

The NFS method disclosed in Abdelnur generates the results and returns it to the client as opposed to the method of claim 1 in which the server returns a set of mounting parameters. The

returned parameters are utilized to automatically mount the volumes. Appellant respectfully submits that returning a result by a server is different than returning a set of mounting parameters and mounting of the volume by the client. Therefore, Appellant respectfully submit that the cited references do not disclose, teach, or even suggest (1) returning from a first server a set of mounting parameters, and (2) automatically mounting a set of volumes by utilizing the returned parameters as recited in claim1.

2. The Cited References Do Not Disclose, Teach, or Suggest a Method that without Receiving a Request from a User of a Client to Mount a Set of Remote Volumes and Requests a Server to Mount the Volumes

In the Office Action, the Examiner correctly specified that Abdelnur does not explicitly teach the method comprising mounting without receiving a request from a user of the client to mount the remote volumes. The Examiner has cited column 5, lines 20-55 of Manwiller for disclosing a system and method for automatically establishing connection with a remote device over a distributed network using API's.

Appellant respectfully submit that Manwiller discloses scheduling a callback function that performs a function of an initialization process that requires a waiting period. *See*, Manwiller, column 2, lines 26-31. Specifically, in the paragraph cited by the Examiner, Manwiller schedules a callback function that is responsible for checking for the completion of an auto negotiation that allows two devices to negotiate common data services. Appellant respectfully submit that the callback function of Manwiller is not a method that without receiving a request from a user requests a server for mounting volumes. Therefore, the combination of the references does not disclose, teach, or even suggest the method of claim 1 that requests a server to mount a set of remote volumes

without receiving a request from a user of a client to mount the volumes.

3. **The Examiner Rejected the Claims by Relying on Impermissible Hindsight to Piecemeal Combine the References**

Appellant respectfully submits that the Examiner's rejection is relying on an improper hindsight and piecemeal reconstruction for rejecting claims 1. The mere fact that references can be combined or modified does not render the resultant combination obvious, unless the prior art also suggests the desirability of the combination. *In re Kotzab*, 217 F.3d 1365, 1371, 55 USPQ2d 1313, 1318 (Fed. Cir. 2000). It is improper to use the inventor's disclosure as a road map for selecting and combining prior art disclosures. *Grain Processing Corp. v. American Maize-Products Corp.*, 840 F.2d 902, 907 (Fed. Cir. 1988). Absent such a showing in the prior art, the Examiner has impermissibly used "hindsight" by using the Appellants' teaching as a blueprint to hunt through the prior art for the claimed elements and combine them as claimed. *In re Zuko*, 111 F.3d 887, 42 USPQ2d 1476 (Fed. Cir. 1997).

The cited references do not suggest the desirability of the combination proposed by the Examiner. The Examiner has impermissibly used hindsight by using the Appellant's own claims as a template for combining features from different references for the claimed elements. The only motivation the Examiner has offered is that the teachings are in the same field of endeavor. Moreover, even if two references are in the same field of endeavor, being in the same field of endeavor by itself does not satisfy the burden of establishing a suggestion or motivation to combine two references. Manwiller discloses a scheduling mechanism in an operating system API, which allows a software driver to specify a function to be called back after a given interval has lapsed. *See*,

Manwiller, column 5, lines 48-51. Abdelnur discloses a method of remote procedure call. The Examiner has not shown how one of ordinary skill will be motivated to combine the references for all elements of claims 1. Therefore, the Examiner has not met the burden of establishing prima facie case of why it makes sense to combine these references.

As indicated above, in the Office Action, the Examiner correctly specified that that Abdelnur does not explicitly teach the method comprising mounting without receiving a request from a user of the client to mount the remote volumes. The Examiner has cited Manwiller for disclosing a system and method for automatically establishing connection with a remote device over a distributed network using API's.

Appellant respectfully submits that the Examiner, in making this rejection, did not identify any suggestion or motivation in the art for combining the two references. Without the identification of a suggestion or motivation to combine the cited references, the Examiner has not met his prima facie case of obviousness. Appellant respectfully submits that a mere disclosure in Manwiller for a method for automatically establishing connection with a remote device is not enough to show that the references suggest that this method can be extended to automatically mount remote volumes to a client. Appellant respectfully submits that the Examiner has rejected claim 1 by relying on impermissible hindsight to piecemeal combine the references.

B. The Subject Claim 23 is Patentable under 35 U.S.C. §103(a) over Abdelnur in View of Manwiller

In rejecting the subject claim 23 under §103, the Examiner stated the following:

As per claim(s) 23 Abdelnur-Manwiller discloses a plug-in module configured to receive from a server a mounting command block, said mounting command block to

include volume mounting parameters, (See Abdelnur, Column 5 Lines 5-40); and an application programming interface coupled to said plug-in and configured to automatically mount said volumes specified in said mounting command block, said mounting performed without the need for user interventions, (See Abdelnur, Column 6 Lines 16-40).

(Office Action mailed March, 25, 2005, page 8)

1. **The Cited References Do Not Disclose, Teach, or Suggest a Plug-in Module that is Configured to Receive a Mounting Command Block that includes Mounting Parameters from a Server**

Appellant respectfully submit that the cited references do not disclose the plug-in module of claim 23 that is configured to receive a mounting command block that includes parameters from a server. Specifically, the cited paragraphs of Abdelnur disclose a Java applet and not a plug-in module:

The classes of a Java applet are loaded on demand from the network (stored on a server), or from a local file system, when first referenced during the Java applet's execution. *See*, Abdelnur column 5, lines 17-19.

For at least two reasons, Appellant respectfully submits that Abdelnur's Java applet is not the recited plug-in module of claim 23. *First*, as is known in the art, plug-in applications are programs that are installed and used as part of a client application (such as a web browser) and are not limited to use the resources of the server from which they are loaded. *Second*, the Java applet disclosed in Abdelnur is not for receiving a mounting command block from the server when the mounting command block includes mounting parameters. Instead, it is for accessing the resources on the server. For instance, column 5, lines 29-34 of Abdelnur states:

A Java applet loaded from the network server is executed on the client's virtual machine. An applet has limited permission to access the resources available on the server and other network computers. In prior art schemes, this access is typically limited to the resources available on the server where the applet is loaded from.

In view of the foregoing, Appellant respectfully submits that the cited references do not disclose, teach, or suggest the apparatus of claim 23 that includes a plug-in module that is configured to receive a mounting command block that includes mounting parameters from a server.

2. **The Cited References Do Not Disclose, Teach, or Suggest an Application Programming Interface Coupled to a Plug-in Module and Configured to Automatically Mount a Set of Remote Volumes Specified in a Mounting Command Block**

Appellant respectfully submit that the cited references do not disclose, teach, or event suggest the application programming interface (API) of claim 23. This API is coupled to a plug-in module and is configured to automatically mount a set of remote volumes specified in a mounting command block. The application programming interface disclosed in Abdelnur is utilized by applications to interface with an NFS client and is not an API connected to a plug-in.

Furthermore, as discussed in Sections A above, the NFS method disclosed in Abdelnur, generates the results and returns it to the client as opposed to the computer system of claim 23 in which (1) the server returns a mounting command block and (2) the application programming interface (that is connected to the plug-in module) automatically mounts the volumes specified in the mounting command block. Appellant respectfully submits that the cited references do not disclose, teach, or even suggest an apparatus that includes an application programming interface coupled to a plug-in module and configured to automatically mount a set of remote volumes specified in a mounting command block.

3. **The Cited References Do Not Disclose, Teach, or Suggest Mounting of Remote Volumes without the Need for User Intervention**

In rejecting claim 23, the Examiner has not identified how the combined references show the

limitation of mounting the remote volumes without the need for user intervention. However, as discussed in Section A above in relation with claim 1, the Examiner has correctly specified that Abdelnur does not explicitly teach the method comprising mounting without receiving a request from a user of the client to mount the remote volumes. The Examiner has cited Manwiller for disclosing a system and method for automatically establishing connection with a remote device over a distributed network using API's.

Appellant respectfully submit that Manwiller discloses scheduling a callback function that performs a function of an initialization process that requires a waiting period. *See*, Manwiller, column 2, lines 26-31. Specifically, in the paragraph cited by the Examiner, Manwiller schedules a callback function that is responsible for checking for the completion of an auto negotiation that allows two devices to negotiate common data services. Appellant respectfully submits that the callback function of Manwiller is not an automatic request to a server for mounting volumes. Therefore, the combination of Abdelnur and Manwiller does not disclose, teach, or even suggest the apparatus of claim 23 that mounts remote volumes without the need for user intervention.

C. The Subject Claim 5 is Patentable under 35 U.S.C. §103(a) over Abdelnur in View of Manwiller

In rejecting the subject claim 5 under §103, the Examiner stated the following:

As per claim(s) 5 Abdelnur-Manwiller teaches the claimed invention as described in claim(s) 1-4 above and further discloses detecting the presence of a cookie on said client, said cookie indicating by its presence that said client has installed a plug-in module; and if said cookie is not detected, terminating any said volume mounting, (See Abdelnur, Column 12 Lines 8-25).

(Office Action mailed March, 25, 2005, page 4)

In addition to be patentable for the reasons given in Section A above, Appellant respectfully submits that claim 5 is also patentable for the following reasons.

1. **The Cited References Do Not Disclose, Teach, or Suggest a Method that Detects the Presence of a Cookie on the Client where the Cookie Indicates by its Presence that the client has Installed a Plug-in Module**

Appellant respectfully submit that the cited references do not disclose, teach, or even suggest the method of claim 5 that detects the presence of a cookie on the client, where the presence of the cookie indicates that the client has installed a plug-in module. Specifically, the cited paragraphs of Abdelnur do not disclose a cookie on the client that indicates the presence of a plug-in module. Instead, the cited paragraphs disclose:

[S]ervlet provides an additional level of security by embedding into a submitted request, identification information or credentials for a client application. Hence, credentials submitted for a client application are ignored and replaced by the ones assigned by servlet. *See*, Abdelnur, column 12, lines 7-12.

Appellant respectfully submits that it is well known in the art that a cookie is a small software code that is placed on the client, as opposed to a servlet, which is a piece of code that is executed by on the server machine. Also, Appellant respectfully submits that the servlet disclosed by Abdelnur is to authenticate an application credential as oppose to a cookie that indicates the presence of a plug-in module in the client. Therefore, Appellant respectfully submits that the cited references do not disclose, teach, or even suggest a method that detects the presence of a cookie on the client where the presence of the cookie indicates that the client has installed a plug-in module

2. The Cited References Do Not Disclose, Teach, or Suggest a Method that Terminates Volume Mounting if a Cookie is not Present on the Client

Appellant respectfully submits that the cited references do not disclose, teach, or even suggest the method of claim 5, which terminates volume mounting if a cookie is not present on the client. Specifically, the servlet identified by the Examiner is not used to terminate volume mounting. Instead, as stated in column 12, lines 7-16 of Abdelnur:

[S]ervlet provides an additional level of security by embedding into a submitted request, identification information or credentials for a client application. Hence, credentials submitted by the client application are ignored and replaced by the ones assigned by servlet. For example, servlet can replace any user credentials submitted by application by a set of identification information previously programmed into servlet. Each user request can be identified by servlet and assigned credentials previously classified for that user. This scheme prohibits malicious client applications from breaching network security by submitting false credentials.

Abdelnur's servlet is, therefore, for replacing user credentials with pre-programmed identification information. As such, this servlet is not for terminating the mounting of remote volumes based on the absence of a cookie that indicates that a plug-in is not installed in the client, as recited in claim 5.

D. The Subject Claim 6 is Patentable under 35 U.S.C. §103(a) over Abdelnur in View of Manwiller

In rejecting the subject claim 6 under §103, the Examiner stated the following:

As per claim(s) 6 Abdelnur-Manwiller teaches the claimed invention as described in claim(s) 1-5 above and furthermore discloses if successfully authenticated, retrieving said parameters from a profile datastore on said first server, said parameters indexed by said login ID and assembling a mounting command block, said command block consisting of a mount command and said retrieved parameters, (See Abdelnur, Column 5 Lines 48-67 & Column 6 Lines 1-21).

(Office Action mailed March, 25, 2005, page 4)

In addition to be patentable for the reasons given in Section A above, Appellant respectfully submits that claim 6 is also patentable for the following reasons.

1. The Cited References Do Not Disclose, Teach, or Suggest Retrieving Mounting Parameters from a Profile Datastore on a Server where the Parameters are Indexed by a login ID

Appellant respectfully submits that the cited references do not disclose, teach, or even suggest the method of claim 6, which retrieves mounting parameters from a profile datastore on a server where the parameters are indexed by a login ID. Specifically, column 6, lines 16-21 of Abdelnur discloses:

The NFS generated mount table is stored as a file on each server and client in the NFS network. The mount table consists of a list of entries. When a new resource is mounted, a new entry is generated in the table automatically and is used to locate the resource whenever a request for access to the resources is made by a client.

The mount table in Abdelnur is, therefore, loaded on both server and client and is used to locate resources that are mounted on the server, as opposed to a profile datastore that retrieves mounting parameters that are particular to a client and are located according to a profile based on the client login ID. There is no disclosure in Abdelnur that a particular login ID would result in a particular set of mounting parameters that are located according to a profile for that particular login ID. accordingly, Appellant respectfully submit that the cited references do not disclose, teach, or even suggest the method of claim 6, which retrieves mounting parameters from a profile datastore on a server where the parameters are indexed by a login ID.

E. The Subject Claim 8 is Patentable under 35 U.S.C. §103(a) over Abdelnur in View of Manwiller

In rejecting the subject claim 8 under §103 the Examiner stated the following:

As per claim(s) 8 Abdelnur-Manwiller teaches the claimed invention as described in claim(s) 1-7 above and furthermore discloses retrieving said parameters from a profile datastore on said first server, said parameters indexed therein by an identifier for said event; and assembling a mounting command block, said command block consisting of a mount command and said retrieved parameters, (See Abdelnur, Column 5 Lines 48-67 & Column 6 Lines 1-21).

(Office Action mailed March, 25, 2005, page 5)

In addition to be patentable for the reasons given in Section A above, Appellant respectfully submits that claim 8 is also patentable for the following reasons.

1. Cited References Do Not Disclose, Teach, or Suggest Retrieving Mounting Parameters from a Profile Datastore on a Server where the Parameters are Indexed by an Event

Appellant respectfully submits that the cited references do not disclose, teach, or even suggest the method of claim 8, which retrieves mounting parameters from a profile datastore on a server where the parameters are indexed by an event. Specifically, the cited column 6, lines 16-21 of Abdelnur discloses:

The NFS generated mount table is stored as a file on each server and client in the NFS network. The mount table consists of a list of entries. When a new resource is mounted, a new entry is generated in the table automatically and is used to locate the resource whenever a request for access to the resources is made by a client.

The mount table in Abdelnur is, therefore, loaded on both server and client and is used to locate resource mounted on the server, as oppose to a profile datastore that retrieves mounting parameters that are particular to a client and are located according to a profile based on an event. Accordingly, Appellant respectfully submit that the cited references do not disclose, teach, or suggest the method of claim 8, which retrieves mounting parameters from a profile datastore on a server where the parameters are indexed by an event.

F. The Subject Claims 9 and 20 are Patentable under 35 U.S.C. §103(a) over Abdelnur in View of Manwiller

In rejecting the subject claim 9 under §103 the Examiner stated the following:

As per claim(s) 9 Abdelnur-Manwiller teaches the claimed invention as described in claim(s) 1-8 above and furthermore discloses said server returns said parameters via a plug-in module installed on said client; wherein a Plug-in can be interpreted as a Java Applet, (See Abdelnur, Column 5 Lines 5-40).

(Office Action mailed March, 25, 2005, page 5)

In addition to be patentable for the reasons given in Section A above, Appellant respectfully submits that claim 9 is also patentable for the following reasons.

1. Cited References Do Not Disclose, Teach, or Suggest the Method of Claim 9 in which the Server Returns the Mounting Parameters via a Plug-in Module Installed on the Client.

Appellant respectfully submits that the cited references do not disclose, teach, or even suggest the method of claim 9 in which the server returns the mounting parameters via a plug-in module installed on the client. Specifically, the cited paragraphs of Abdelnur disclose a Java applet and not a plug-in module. for instance, column 5, lines 17-19 of Abdelnur states:

The classes of a Java applet are loaded on demand from the network (stored on a server), or from a local file system, when first referenced during the Java applet's execution.

For at least two reasons, Appellant respectfully submits that Abdelnur's Java applet is not the recited plug-in module of claim 9. *First*, as is known in the art, plug-in applications are programs that are installed and used as part of a client application (such as a web browser) and are not limited to use the resources of the server from which they are loaded. *Second*, the Java applet disclosed in

Abdelnur is not for receiving a mounting command block from the server when the mounting command block includes mounting parameters. Instead, it is for accessing the resources on the server. For instance, column 5, lines 29-34 of Abdelnur states:

A Java applet loaded from the network server is executed on the client's virtual machine. An applet has limited permission to access the resources available on the server and other network computers. In prior art schemes, this access is typically limited to the resources available on the server where the applet is loaded from.

In view of the foregoing, Appellant respectfully submits that the cited references do not disclose, teach, or suggest the method of claim 9 in which the server returns the mounting parameters via a plug-in module installed on the client.

G. The Subject Claim 10 is Patentable under 35 U.S.C. §103(a) over Abdelnur in View of Manwiller

In rejecting the subject claim 10 under §103, the Examiner stated the following:

As per claim(s) 10 Abdelnur-Manwiller teaches the claimed invention as described in claim(s) 1-9 above and furthermore discloses obtaining an address for said first server; comparing said address for said first server against a built-in list of approved servers; and if said address is not contained on said built-in list, then terminating the process of automatically mounting; wherein obtaining an address is inherent in that when a selection is made for a certain NFS server the only way to obtain that specific server is by an address of some sort, and if an address does not exist then no mounting occurs as a result of not finding a server specific to the address obtained, (See Abdelnur, Column 7 Lines 47-63 & Column 8 Lines 36-52).

(Office Action mailed March, 25, 2005, page 6)

In addition to be patentable for the reasons given in Sections A and F above, Appellant respectfully submits that claim 10 is also patentable for the following reasons.

1. Cited References Do Not Disclose, Teach, or Suggest the Method of Claim 10 that Compares the Address for the Server Against a Built-in

List of Approved Servers

Appellant respectfully submits that the cited references do not disclose, teach, or even suggest the method of claim 10, which compares the address for the server against a built-in list of approved servers. Specifically, the cited paragraphs in Abdelnur disclose that when the NFS server finds a file that is requested by an application, the NFS server verifies whether the requesting application has permission to access the file system. *See* Abdelnur column 7, lines 49-52.

Appellant respectfully submit that the NSF server that verifies whether an application has permission to access a file is different than comparing the address of the server against a built-in list of approved servers. Therefore, the cited references do not disclose, teach, or even suggest the method of claim 10, which compares the address for the server against a built-in list of approved servers.

H. The Subject Claims 11 and 21 are Patentable under 35 U.S.C. §103(a) over Abdelnur in View of Manwiller

In rejecting the subject claim 11 under §103, the Examiner stated the following:

As per claim(s) 11 Abdelnur-Manwiller teaches the claimed invention as described in claim(s) 1-10 above and furthermore discloses passing of said parameters by said plug-in to an application programming interface (API) of said client; and completing the mounting of said volume by said API, said mounting achieved absent of user intervention, (See Abdelnur, Column 6 Lines 15-40).

(Office Action mailed March, 25, 2005, page 6)

In addition to be patentable for the reasons given in Sections A, F, and G above, Appellant respectfully submits that claim 11 is also patentable for the following reasons.

1. **The Cited References Do Not Disclose, Teach, or Suggest an Application Programming Interface that receives a Set of Mounting Parameters from a Plug-in Module and Mounts a Set of Remote Volumes**

Appellant respectfully submit that the cited references do not disclose, teach, or event suggest the application programming interface of claim 11 that receives a set of mounting parameters from a plug-in module and mounts a set of remote volumes. Specifically, the application programming interface (API) indicated in Abdelnur is utilized by applications to interface with an NFS client and is not an API connected to a plug-in.

Furthermore, as discussed in Sections A above, the NFS method disclosed in Abdelnur, generates the results and returns it to the client as opposed to the method of claim 11, in which (1) the application programming interface receives the mounting parameters from a plug-in module and (2) the application programming interface mounts the volumes. Appellant respectfully submits that the cited references do not disclose, teach, or even suggest the application programming interface of claim.

2. **The Cited References Do Not Disclose, Teach, or Suggest Mounting of Remote Volumes Absent User Intervention**

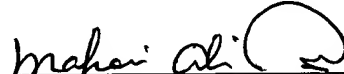
In rejecting claim 11, the Examiner has not identified how the combined references show the limitation of mounting the remote volumes without the need for user intervention. However, as discussed in Section A above in relation with claim 1, the Examiner has correctly specified that Abdelnur does not explicitly teach the method comprising mounting without receiving a request from a user of the client to mount the remote volumes. The Examiner has cited Manwiller for disclosing a system and method for automatically establishing connection with a remote device over

a distributed network using API's.

Appellant respectfully submit that Manwiller discloses scheduling a callback function that performs a function of an initialization process that requires a waiting period. *See*, Manwiller, column 2, lines 26-31. Specifically, in the paragraph cited by the Examiner, Manwiller schedules a callback function that is responsible for checking for the completion of an auto negotiation that allows two devices to negotiate common data services. Appellant respectfully submit that the callback function of Manwiller is not a method that without receiving a request from a user requests a server for mounting volumes. Therefore, the combination of Abdelnur and Manwiller does not disclose, teach, or even suggest the method of claim 11 that mounts remote volumes absent the user intervention.

CONCLUSION

In view of the foregoing, Appellants respectfully submit that the appealed claims, namely claims 1-21 and 23-24 are patentable. Appellants hereby request that the Board overturn the Examiner's finding that the claims are unpatentable under 35 U.S.C. 103(a).

BY: 
Ali Makoui
Reg. No. 45,536
Date: December 29, 2005
Tel. No.: 310-785-0140 ext. 302

12/29/05

CLAIMS APPENDIX

The following claims are the subject of this Appeal.

1. A method for automatically mounting a plurality of remote volumes to a client, comprising:

without receiving a request from a user of said client to mount said plurality of remote volumes, requesting of a first server said mounting of said volumes;

returning by said first server a set of mounting parameters; and

automatically mounting said plurality of volumes utilizing said returned parameters, said volumes after mounting behaving as native to said client, even though said volumes are located remote to said client.
2. A method according to claim 1 wherein said requesting includes:

authenticating said client by said first server via an interface to said first server, said client authentication characterized by a login identification (ID).
3. A method according to claim 2 wherein said authenticating includes:

supplying by the client said login ID and a password intended for said login ID.
4. A method according to claim 3 wherein said authenticating further includes:

comparing said supplied password and said login ID with a stored password on said first server associated with said login ID; and

if said stored password matches said supplied password for said login ID, then characterizing said client as successfully authenticated with said login ID.

5. A method according to claim 1 that further includes:
detecting the presence of a cookie on said client, said cookie indicating by its presence that said client has installed a plug-in module; and
if said cookie is not detected, terminating any said volume mounting.
6. A method according to claim 4 wherein said returning includes:
if successfully authenticated, retrieving said parameters from a profile datastore on said first server, said parameters indexed therein by said login ID; and
assembling a mounting command block, said command block consisting of a mount command and said retrieved parameters.
7. A method according to claim 1 wherein said requesting is initiated by an event occurring on said client requiring said mounting.
8. A method according to claim 7 wherein said returning includes:
retrieving said parameters from a profile datastore on said first server, said parameters indexed therein by an identifier for said event; and
assembling a mounting command block, said command block consisting of a mount command and said retrieved parameters.
9. A method according to claim 1 wherein said first server returns said parameters via a plug-in module installed on said client.
10. A method according to claim 9 wherein automatically mounting includes:
obtaining an address for said first server;
comparing said address for said first server against a built-in list of approved servers; and
if said address is not contained on said built-in list, then terminating the process of automatically mounting.

11. A method according to claim 10 further wherein if said address is contained on said built-in list, then:

passing of said parameters by said plug-in to an application programming interface (API) of said client; and

completing the mounting of said volume by said API, said mounting achieved absent of user intervention.

12. A method according to claim 1 wherein said remote interface is an Hypertext Transport Protocol capable browser.

13. A method according to claim 1 wherein said parameters include, for each of said volumes, a volume name, a server alias, a volume username, and a volume password.

14. A method according to claim 13 wherein said server alias is resolved to an address of a second server, said second server housing the volume identified by the corresponding said volume name.

15. A method according to claim 14 wherein said address of said second server is an Internet Protocol (IP) address.

16. A method according to claim 14 wherein said resolving is achieved by:
looking up said server alias in a mapping table; and
retrieving from said look up said address of said server from said mapping table.

17. A method according to claim 6 wherein said mounting command block is contained in an extensible markup language (XML) document.

18. A method according to claim 8 wherein said mounting command block is contained in an extensible markup language (XML) document.

19. A system for automatically mounting volumes over a network, said system

comprising:

a client configured to automatically request said mounting of said volumes without the need for a user of said client to request said mounting of said volumes; and

a server coupled to said client, said server configured to return a mounting command block to said client.

20. A system according to claim 19 further comprising:

a plug-in installed in said client, said plug-in configured to approve the address of said server.

21. A system according to claim 20 further comprising:

an application programming interface (API) coupled to said plug-in and configured to automatically mount said volumes specified in said mounting command block, without the need for user intervention.

22. (Canceled)

23. In a computer system having a processor and memory, an apparatus for automatically mounting a plurality of remote volumes to said computer system, said apparatus comprising:

a plug-in module configured to receive from a server a mounting command block, said mounting command block comprising volume mounting parameters; and

an application programming interface coupled to said plug-in module and configured to automatically mount said volumes specified in said mounting command block, said mounting performed without the need for user intervention.

24. An apparatus for automatically mounting a plurality of volumes to a client, said apparatus comprising:

means for requesting of a first server said mounting of said volumes without receiving a request from a user of said client to mount said plurality of volumes;

means for returning by said first server a set of mounting parameters; and

means for automatically mounting said plurality of volumes utilizing said returned parameters, said volumes after mounting behaving as native to said client, even though said volumes are located remote to said client.

EVIDENCE APPENDIX

There is no evidence submitted by Appellants, the Appellants' legal representative, or assignees thereof.

RELATED PROCEEDINGS APPENDIX

There are no related appeals or interferences known to Appellants, the Appellants' legal representative, or assignees thereof.